Feasible Real-time Nonlinear Model Predictive Control

Matthew J. Tenny

September 27, 2000

Contents

- Nonlinear Model Predictive Control
  - Process Model
  - Motivating Example
  - Controller Formulation
  - Suboptimal Control
- Feasibility
  - Input Constraints
  - State Constraints
  - Framework for Suboptimal NMPC
- Example Revisited
- Copolymerization Example
- Conclusions and Future Work

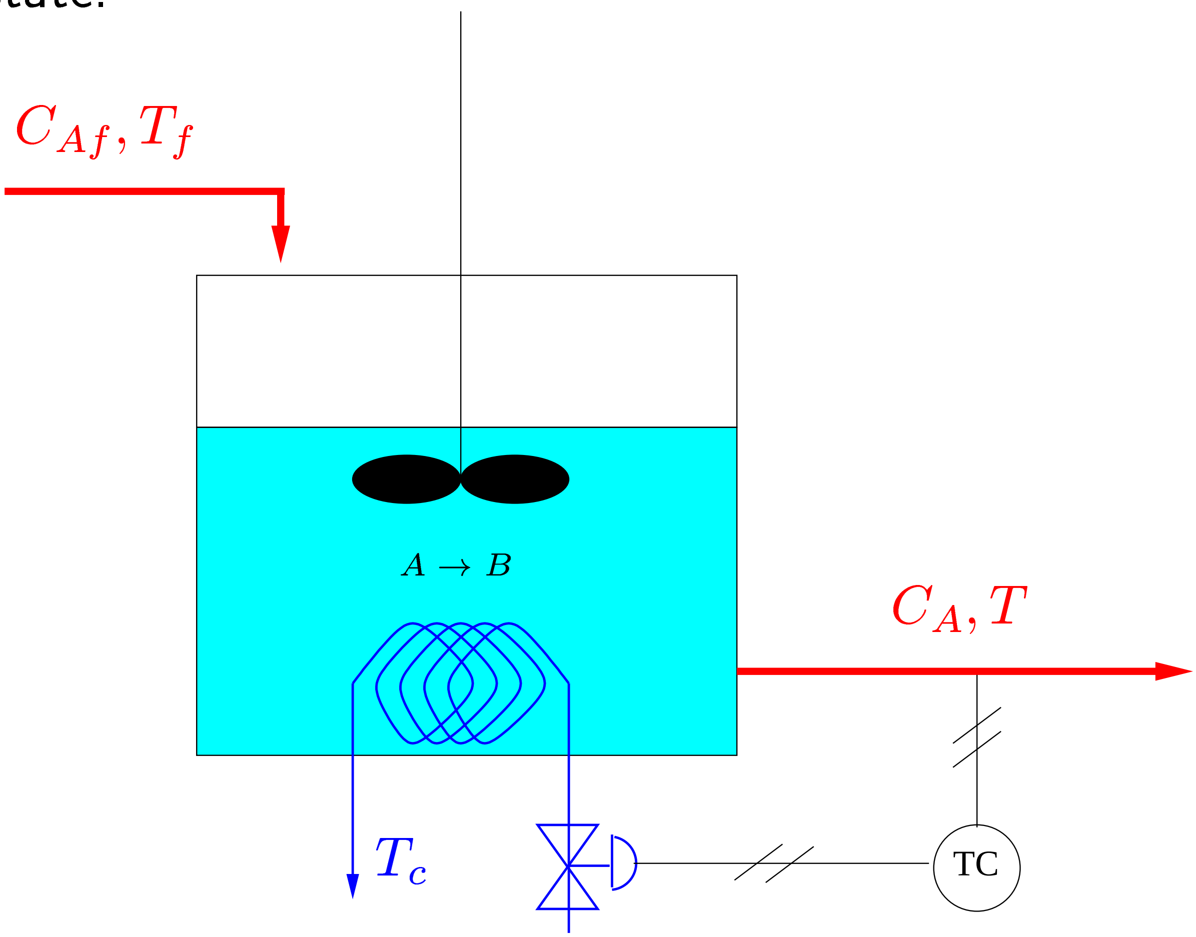
Why Use Nonlinear MPC?

- Nonlinear dynamics are ubiquitous in chemical engineering applications
  - Conservation laws: product of flowrates and concentration or enthalpy
  - Kinetics: nonlinear rate laws, temperature dependence of rate constants
  - Thermodynamics: vapor/liquid equilibrium

- Non-quadratic cost function and nonlinear constraints might be more natural for an application
  - optimize endpoint of batch run, minimize off-spec production in a grade transition
  - optimize startup or shutdown procedure
  - minimize waste, minimize cost, maximize efficiency, maximize yield
  - product properties are nonlinear functions of state, e.g. polymer viscosity as a function of MWD

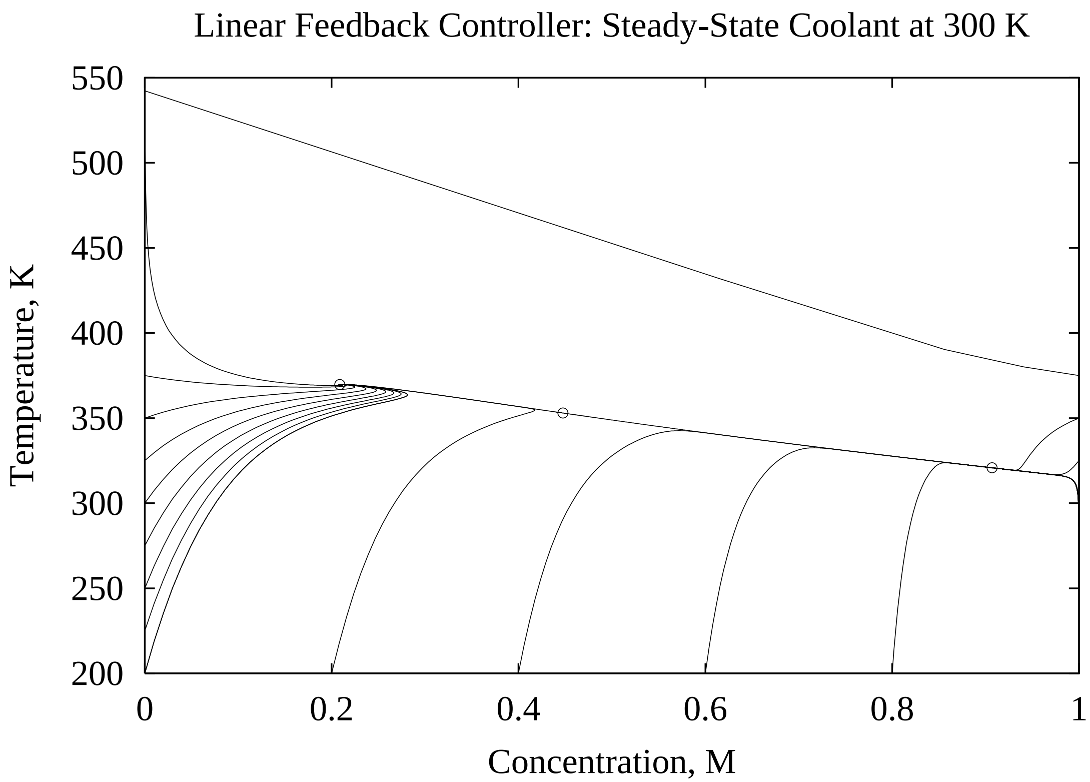
Motivating Example

We adapt the simple model presented by Henson and Seborg [1] for a continuously stirred tank reactor (CSTR) undergoing reaction $A \rightarrow B$ at an unstable steady state.



[1]M. A. Henson and D. E. Seborg. *Nonlinear Process Control.* Prentice Hall PTR, Upper Saddle River, New Jersey, 1997.

Example Under Linear Control



Nonlinear Model

Continuous Time:

$$\frac{dx}{dt} = f(x, u)$$

$$y = g(x)$$

$$u \in U$$

$$x \in X$$

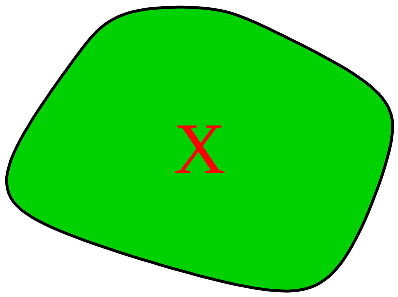
Discrete Time:

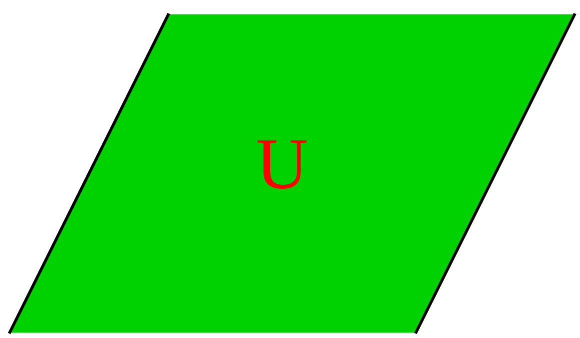
$$x_{j+1} = F(x_j, u_j)$$

$$y_j = g(x_j)$$

$$u_j \in U$$

$$x_j \in X$$





Model Predictive Control Formulation

Infinite Horizon:

$$\min_{u^N} \Phi_k = \sum_{j=0}^{\infty} L(x_{k+j}, u_{k+j})$$

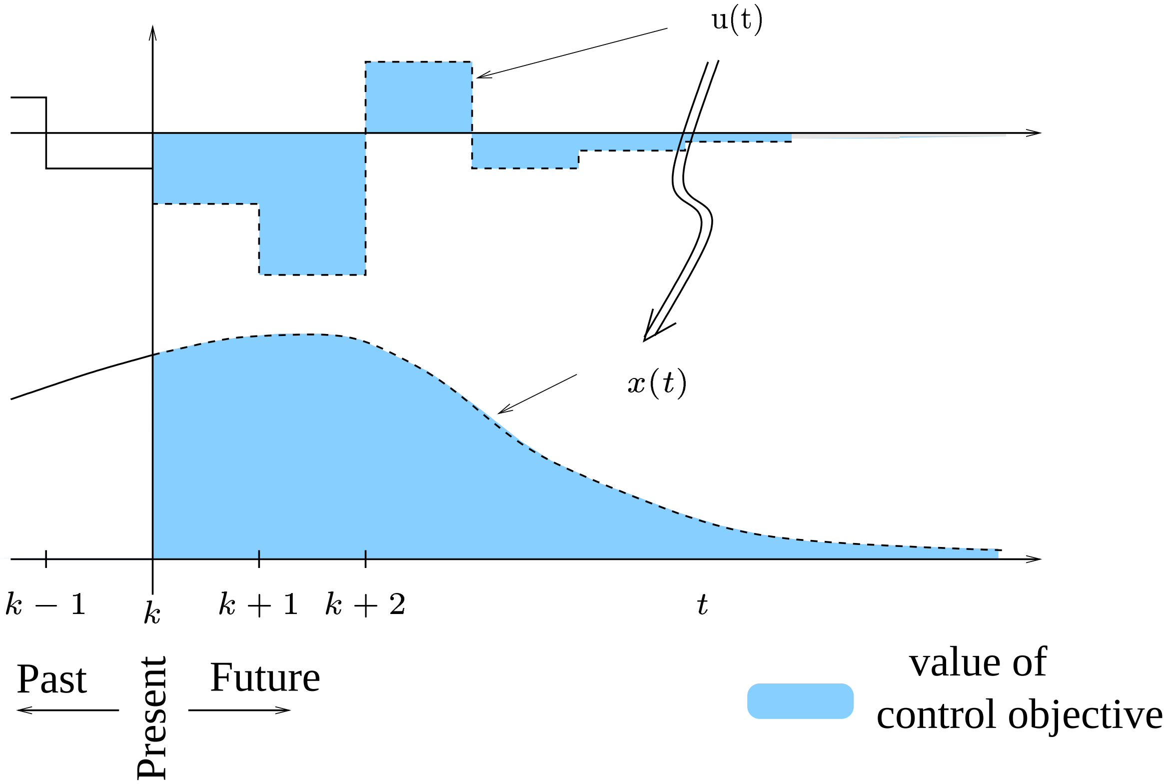
Finite Horizon:

$$\min_{u^N} \Phi_k = \sum_{j=0}^{N} L(x_{k+j}, u_{k+j})$$

with the additional constraint $x_N = 0$.

Commonly, $L(x, u)$ is quadratic.

Model Predictive Control Formulation



Challenges of Standard MPC Scheme

- Computation of a global optimum for the cost function is not always fast or reliable.
- The endpoint stability constraint complicates the optimization problem.
- Poor initial guesses may prevent optimization algorithm from making progress.

Proposed Solution: Suboptimal MPC

- Rather than finding the optimal sequence of control moves, we settle for a feasible sequence of moves that decrease the cost function from its prior value.

- Most of the computation to determine the endpoint region and the initial control sequence can be performed off-line to increase speed of performance.

Suboptimal MPC:

$$\min_{u^N} \Phi_k = \sum_{j=0}^{N-1} (x_{k+j}^T Q x_{k+j} + u_{k+j}^T R u_{k+j} + \Delta u_{k+j}^T S \Delta u_{k+j}) + x_N^T \Pi x_N$$

with the additional constraint

$$x_N \in W_\alpha$$

What is Feasibility?

A sequence of controls is feasible if

- All control moves satisfy the input constraints $u \in U$.
- All states (nearly) satisfy the state constraints $x \in X$.
- The states and inputs satisfy the model equation $x_{j+1} = F(x_j, u_j)$.

Standard optimization methods do not maintain feasibility at each major iteration.

Time-Varying Linearized Model

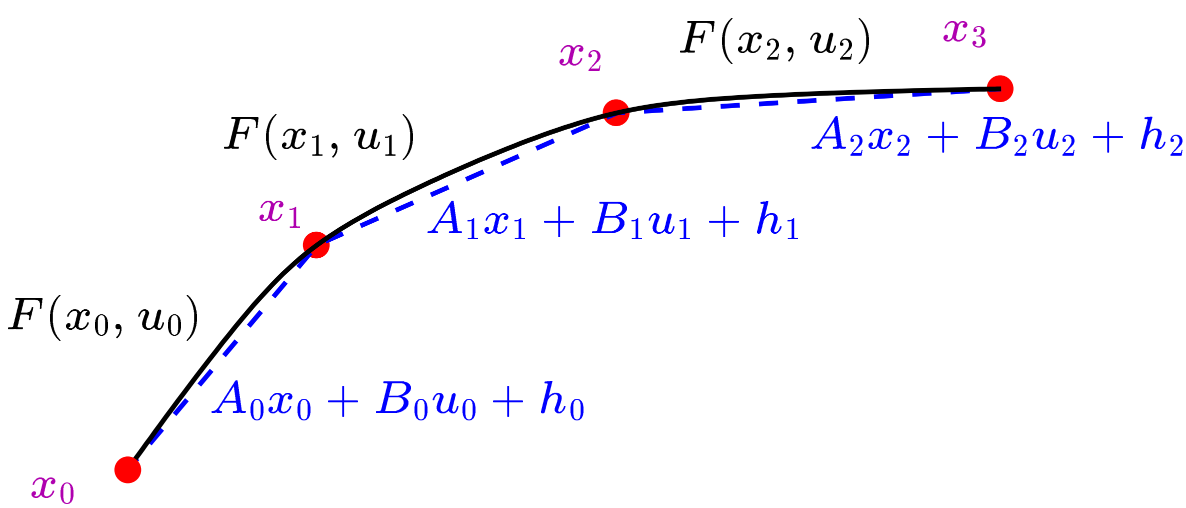
We can linearize our discrete model about each state and input (x_k, u_k) to yield the following form:

$$x_{k+1} = A_k x_k + B_k u_k + h_k$$

Where

$$\begin{aligned} A_k &= \frac{\partial F}{\partial x_k}(\mathbf{x_k}, \mathbf{u_k}) \\ B_k &= \frac{\partial F}{\partial u_k}(\mathbf{x_k}, \mathbf{u_k}) \\ h_k &= F(x_k, u_k) - A_k x_k + B_k u_k \end{aligned}$$

using a discretization time Δt.



QP Subproblem

Solve the problem as a sequence of quadratic programs (SQP).

$$\min_{x,u} \Phi = \sum_{k=0}^{N} \left[x_k^T Q_k x_k + u_k^T R_k u_k \right] + x_N^T \Pi x_N$$

subject to:

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + h_k \\ x_k \in X, & \qquad u_k \in U \end{aligned}$$

Question:

Why formulate the problem with equality constraints?

Banded Lagrangian

$$
\begin{bmatrix}
R_0 & B_0^T & & & & & & & & & & & \\
B_0 & 0 & -I & & & & & & & & & & \\
 & -I & Q_1 & M_1 & A_1^T & & & & & & & & \\
 & & M_1^T & R_1 & B_1^T & & & & & & & & \\
 & & A_1 & B_1 & 0 & -I & & & & & & & \\
 & & & & -I & Q_2 & M_2 & A_2^T & & & & & \\
 & & & & & & \ddots & & & & & & \\
 & & & & & & M_{N-2}^T & R_{N-2} & B_{N-2}^T & & & & \\
 & & & & & & A_{N-2} & B_{N-2} & 0 & -I & & & \\
 & & & & & & & & -I & Q_{N-1} & M_{N-1} & A_{N-1}^T & \\
 & & & & & & & & & M_{N-1}^T & R_{N-1} & B_{N-1}^T & \\
 & & & & & & & & & A_{N-1} & B_{N-1} & 0 & -I \\
 & & & & & & & & & & & -I & \Pi
\end{bmatrix}
\begin{bmatrix}
u_0 \\ \lambda_1 \\ x_1 \\ u_1 \\ \lambda_2 \\ x_2 \\ \vdots \\ u_{N-2} \\ \lambda_{N-1} \\ x_{N-1} \\ u_{N-1} \\ \lambda_N \\ x_N
\end{bmatrix}
=
\begin{bmatrix}
r_0^u \\ r_0^p \\ r_1^x \\ r_1^u \\ r_1^p \\ r_2^x \\ \vdots \\ r_{N-2}^u \\ r_{N-2}^p \\ r_{N-1}^x \\ r_{N-1}^u \\ r_{N-1}^p \\ r_N^x
\end{bmatrix}
$$

(Rao, Wright, Rawlings; JOTA, 1998)

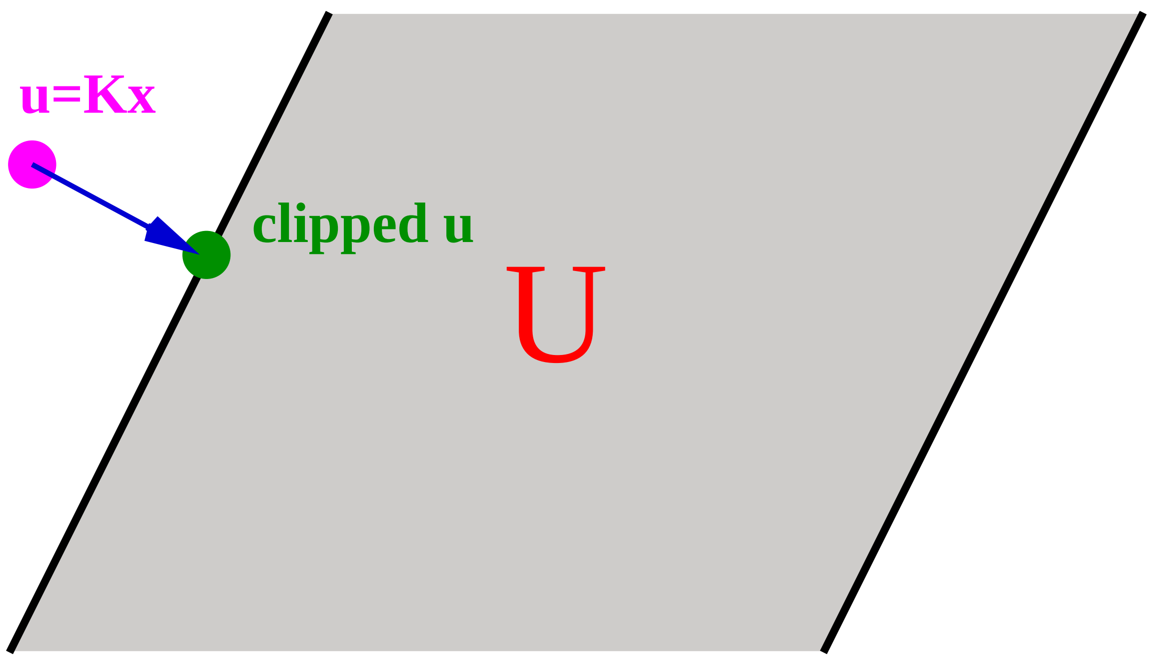
Terminal Constraint

We exclude the terminal region constraint in the quadratic program. This is done for a number of reasons:

- If the horizon length N is chosen to be too short, the problem is infeasible.
- If the horizon length is barely long enough to reach the terminal region, the closed-loop solution may differ appreciably from the open-loop prediction.
- Ellipsoidal constraints in quadratic programs are not exact; the constraint would be approximated.

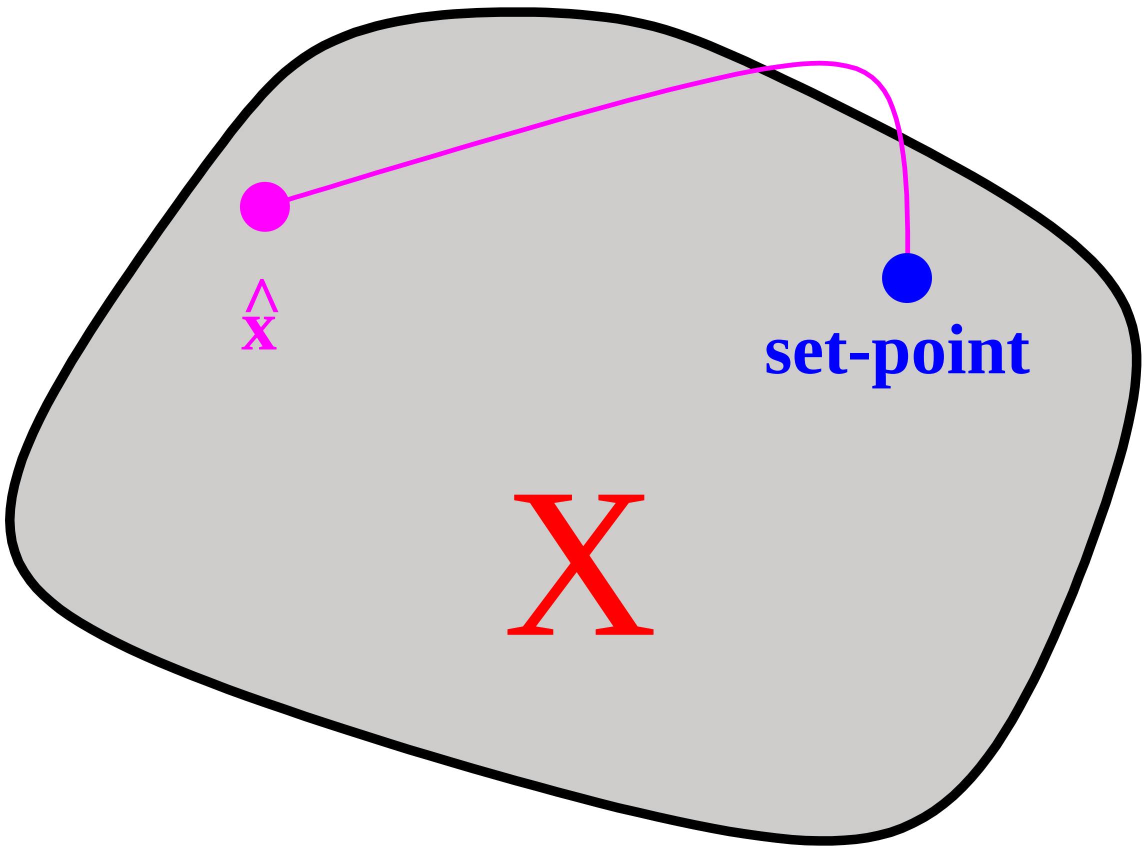
Input Constraints

Generate feasible sequence using feedback law $u_k = Kx_k$. If $u_k \notin U$, then we clip u_k:



State Constraints

We treat the state constraints $x \in X$ as soft constraints. The optimizer is allowed to violate these constraints, but penalizes such violations.



Maintaining Feasibility

The quadratic program yields a new input and state

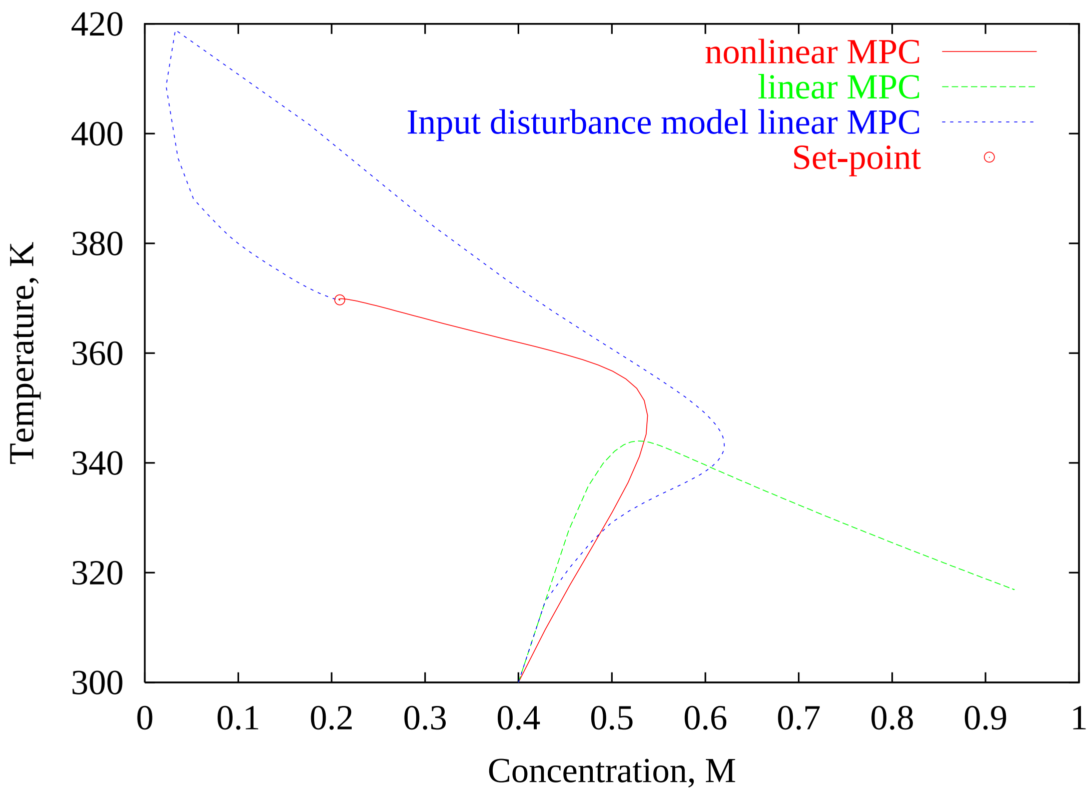
$$\text{QP} \longrightarrow u^{\text{QP}}, x^{\text{QP}}$$

Problem: The solution violates the model

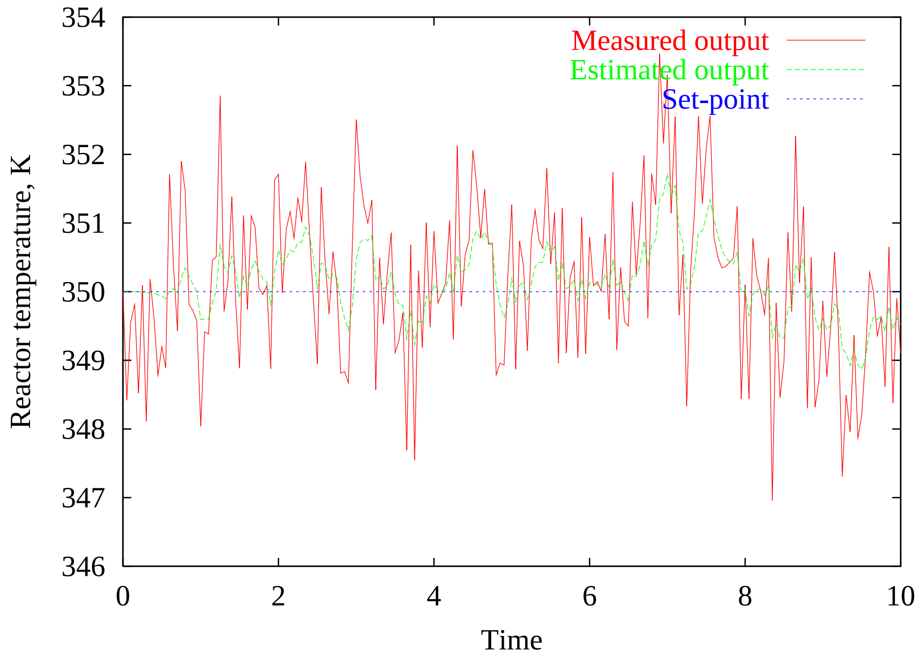
$$x_{k+1}^{\text{QP}} \neq F(x_k^{\text{QP}}, u_k^{\text{QP}})$$

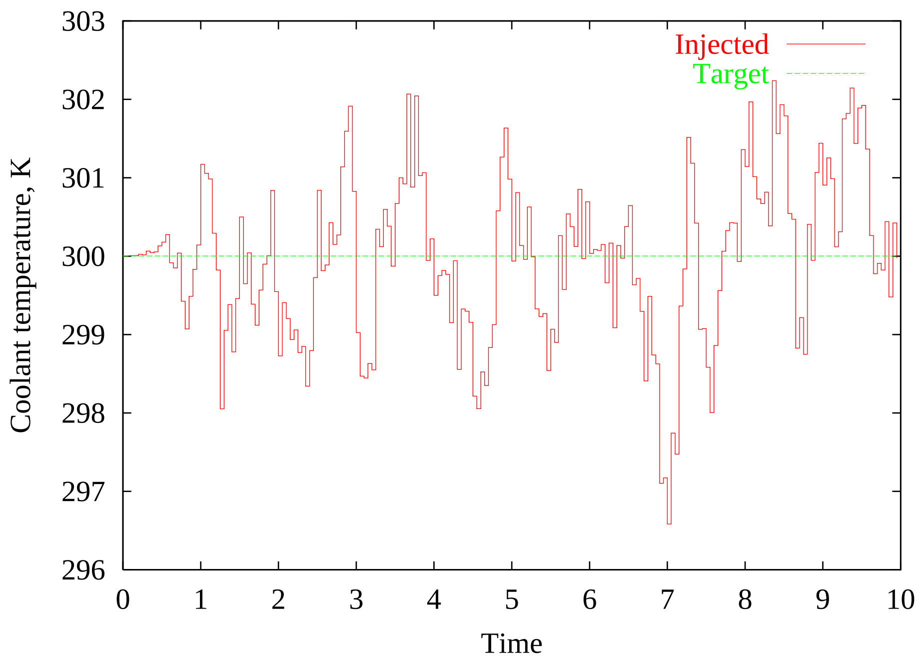
Answer: Substitute u^{QP} into model and generate new states. We check the cost function of the new states, including soft constraint penalties. If the cost is lower, we accept the new inputs. Otherwise, we refine the trust region or line search method.

Example Revisited

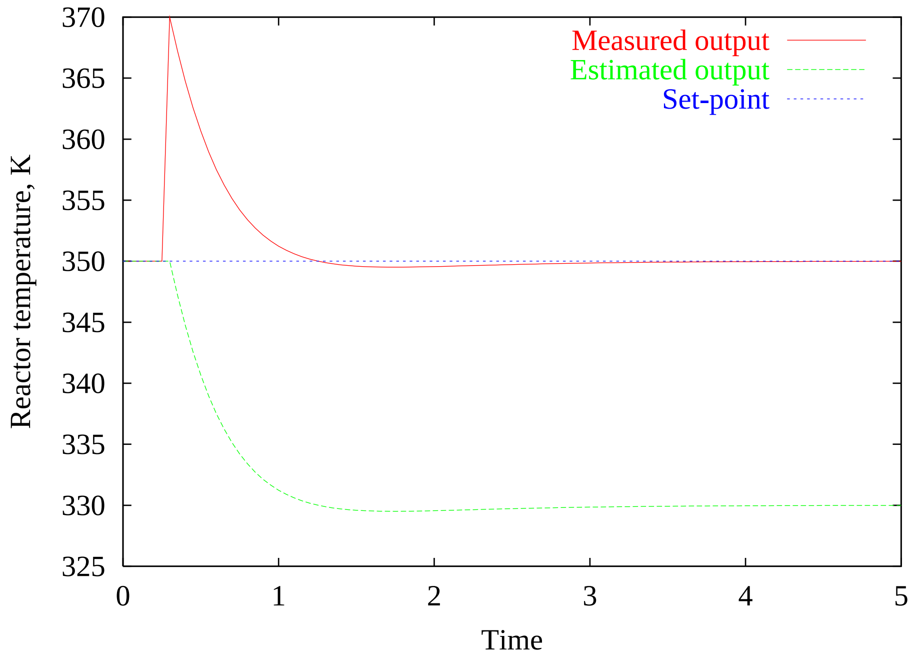


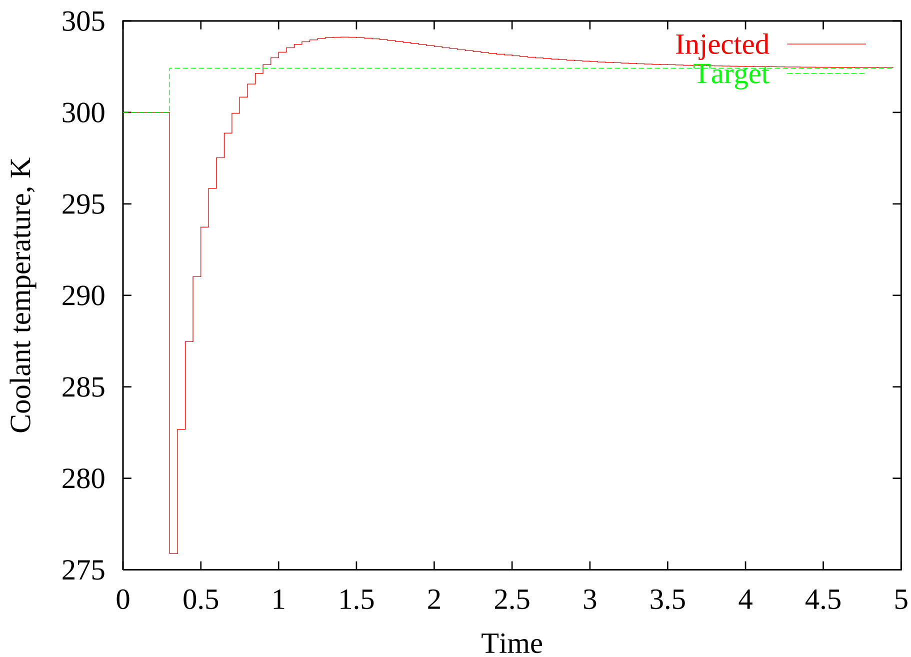
Noise Rejection



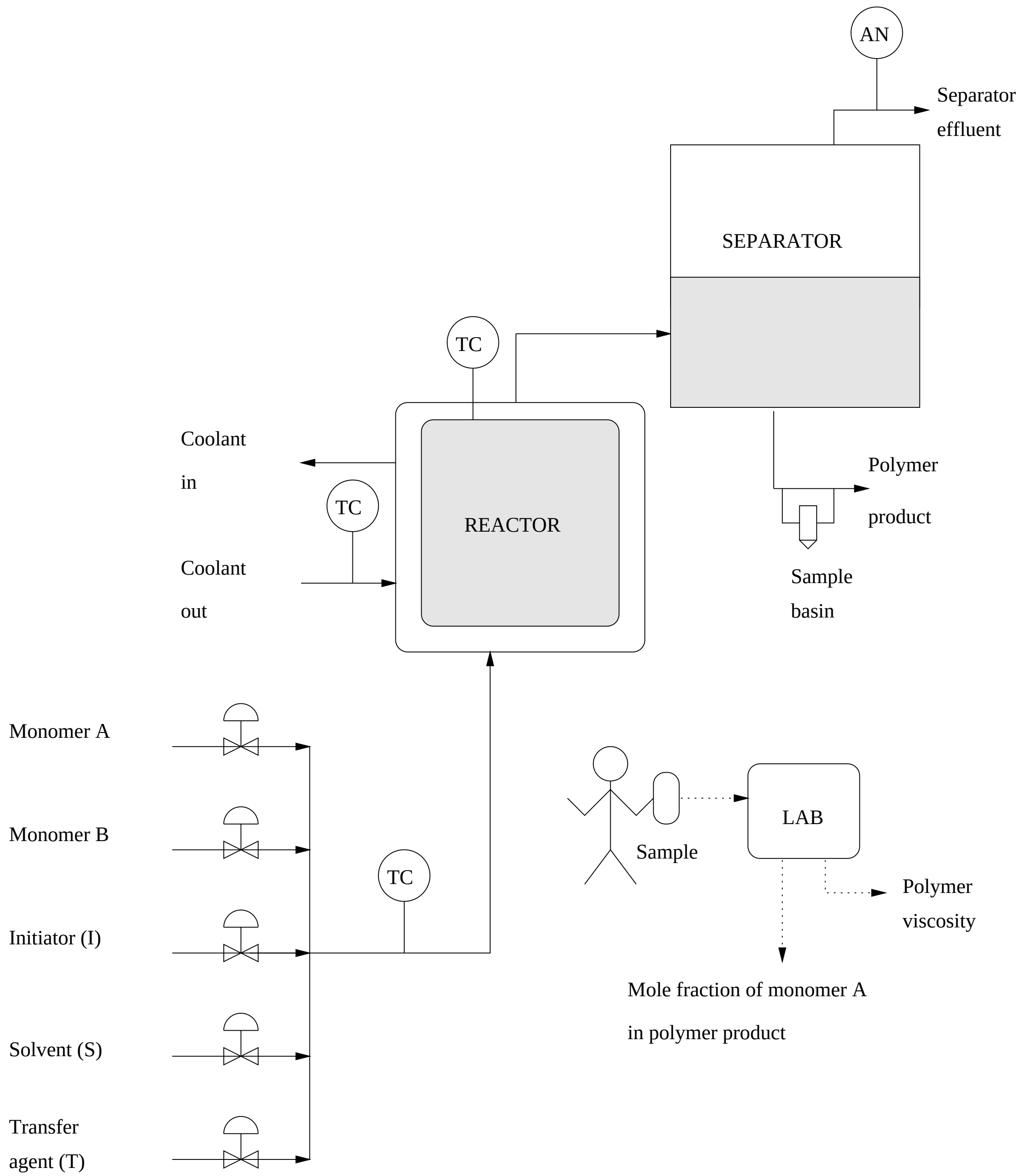


Rejection of Step Disturbance





Copolymerization Example

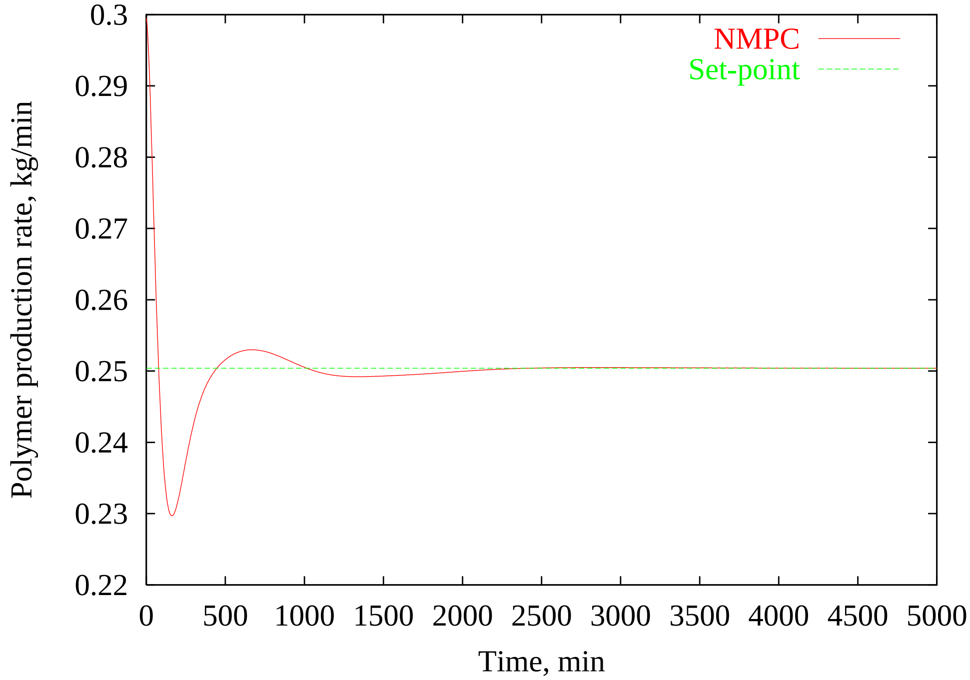


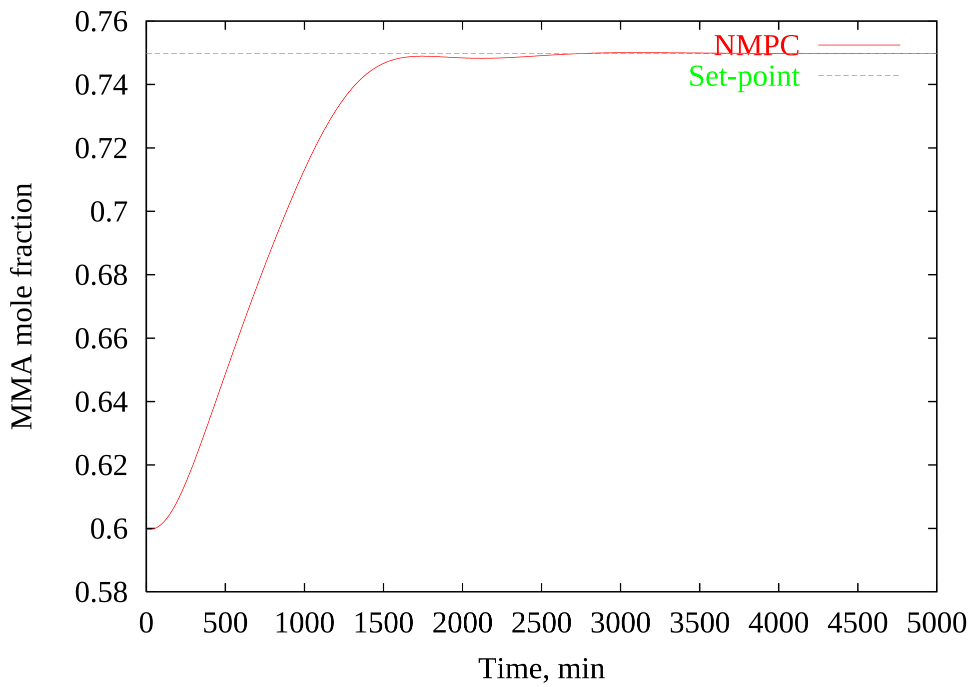
Copolymerization Example

Goal: Transition from Grade A to Grade B

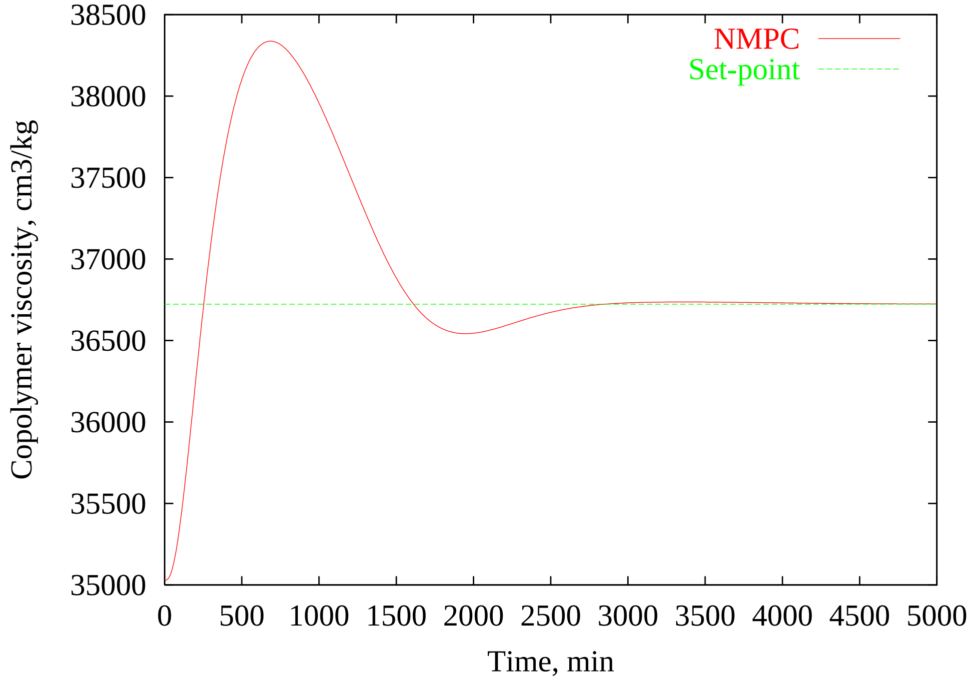
Grade	Polymer production rate (kg/min)	MMA Mole fraction in copolymer	Copolymer viscosity (10^{-6} m^3/kg)
A	0.3	0.60	35000
B	0.25	0.75	36725

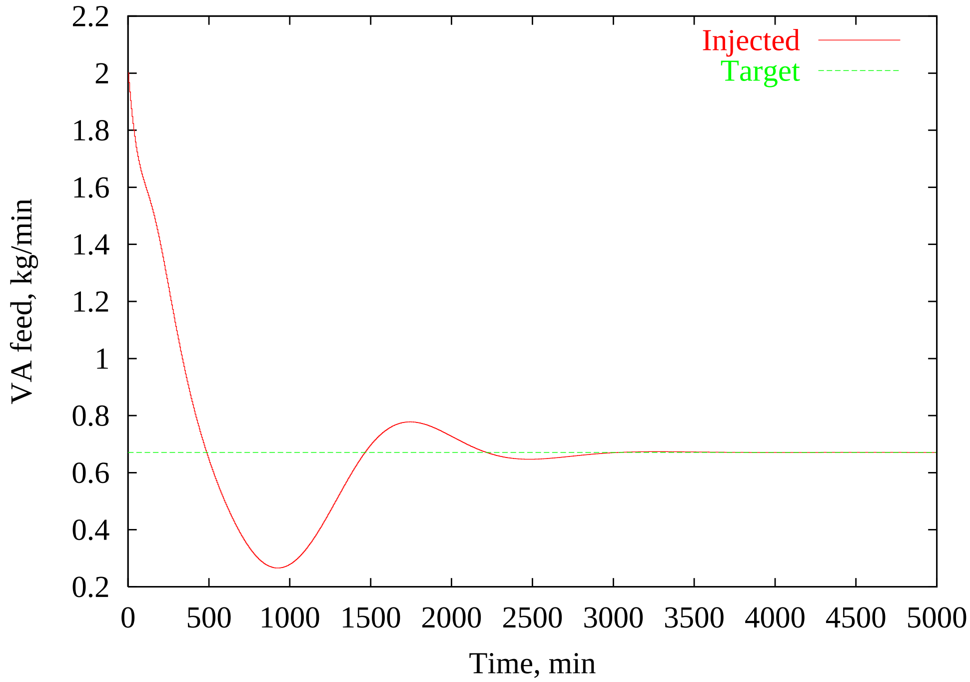
Copolymerization Outputs



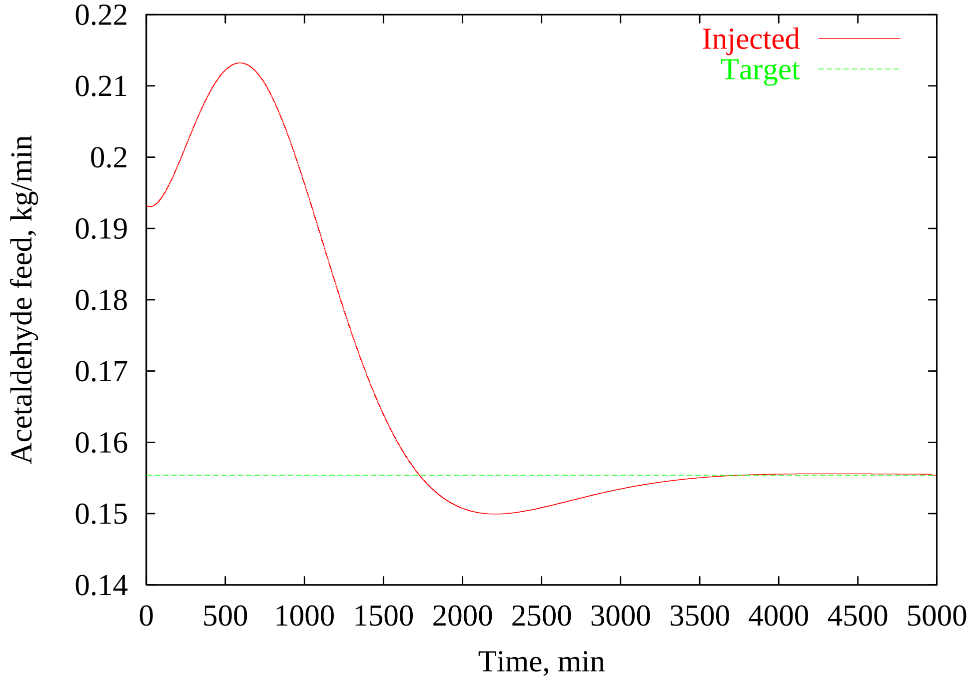


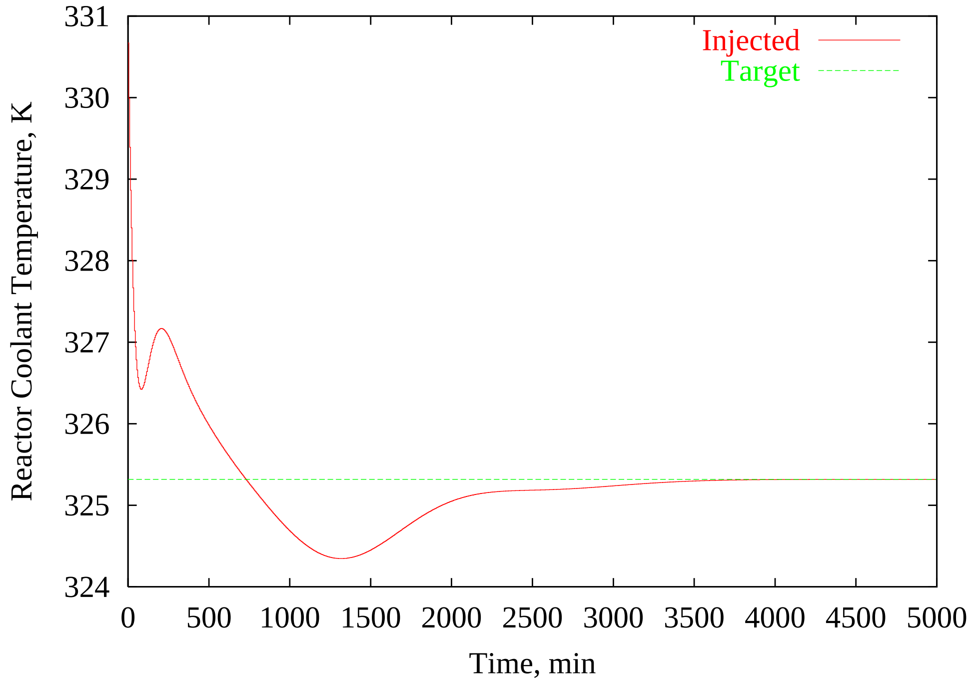
Copolymerization Outputs, Inputs





Copolymerization Inputs





Conclusions & Future Work

Conclusions:

- **For the first time**, we have a nonlinear MPC algorithm we feel is robust against failure and is implementable in real-time.

- NMPC performance is superior to linear control in several kinds of relevant industrial processes.

Future Work:

- Structured moving horizon estimation to incorporate constraints in state estimator

- Incorporation of second derivative information to speed convergence

- Practical application without terminal region

Acknowledgments

- Dr. Stephen J. Wright, MCS Division, Argonne National Labs
- Dr. Rahul Bindlish, Dow Chemical